System and Method for Processing Caller Information Across Heterogeneous Networks

EL977165986US

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates in general to a system and method for processing caller information across heterogeneous networks. More particularly, the present invention relates to a system and method for receiving caller attributes over a computer network and processing an incoming telephone call using the preferred caller attributes.

2. Description of the Related Art

Caller identification (caller ID) became possible with the implementation of Signaling System 7 (SS7) and is a technique to include a caller's telephone number in a telephone call to a call recipient. SS7 provides a signaling backbone for the Public Switched Telephone Network (PSTN) which transfers call information from one central office to another central office. With the implementation of SS7, it became practical to forward a caller's telephone number through the SS7 network to a central office serving a call recipient (i.e. terminating central office).

A terminating central office receives a caller's phone number, and embeds the phone number in a telephone call to the call recipient between the first and second ring of the telephone call. In some instances, a name associated with the initiating calling number is included in the

transmission message. The information is sent in one of two formats which are a Single Data Message Format (SDMF) and a Multiple Data Message Format (MDMF). The SDMF includes the date, time, and a caller's telephone number. The MDMF includes the date, time, caller's telephone number, and a name associated with the caller's telephone number.

A business's call center may use caller ID information in order to access a customer's profile. For example, a call center may use the caller's telephone number to retrieve the caller's address and shopping history from the business's local database. A challenge found, however, is that a call center's database may be outdated and, therefore, not valid. Using the example described above, the customer may have moved to a new residence while keeping his same telephone number. In this example, the business's local database includes an outdated address corresponding to the customer's telephone number.

In addition, a call center may prefer to receive caller information other than what is provided by caller ID to process a corresponding call. Using the example described above, the call center prefers to receive the caller's address but, however, the call center is required to maintain a database to look-up a caller's address because the caller's address is not provided with existing caller ID information.

What is needed, therefore, is a system and method to receive preferred and accurate caller information and process an incoming telephone call using the caller information.

SUMMARY

It has been discovered that the aforementioned challenges are resolved by processing an incoming telephone call using accurate, preferred caller attributes received from a name resolution adapter. The name resolution adapter uses a call recipient agreement corresponding to an enterprise application in order to identify the enterprise application's preferred caller attributes. The name resolution adapter retrieves caller attributes from an accurate database, and includes the caller attributes in a message. The enterprise application receives the message and processes a corresponding call using the caller attributes.

An initiating caller places a call that is intended for an enterprise application, such as one that supports a department store's call center. The initiating caller's switch sends the call over a synchronous optical network (SONET) to a terminating switch that supports the enterprise application. In addition, the initiating caller's switch sends the caller's telephone number to the terminating switch over a signaling system 7 (SS7) network. The terminating switch sends a message to a name resolution adapter whereby the name resolution adapter retrieves one or more caller attributes corresponding to the initiating caller. Once the name resolution adapter is finished retrieving the caller attributes, the name resolution adapter includes the caller attributes, along with port location information, in a message, and sends the message to the enterprise application over a computer network, such as a TCP/IP network.

The enterprise application receives the message and extracts the caller attributes from the message. The enterprise application uses one of the caller attributes to retrieve a caller profile and determine whether to accept the call based upon the caller profile. For example, the enterprise application may support a retail store call center whereby the enterprise application retrieves a caller's account history. In this example, if the caller is not an existing customer of the retail store, the enterprise application may not accept the call in order to not incur long distance charges.

Once the enterprise application decides to accept the call, the enterprise application extracts the port location information from the message. The port location information includes the enterprise application's port location (i.e. circuit) that the call is arriving. The enterprise application detects the call at the port location, and retrieves service subscriptions corresponding to the initiating caller. For example, a caller's service subscriptions may allow a caller to review his billing history, but may not allow him to purchase more merchandise over the telephone because he has reached his credit limit.

The enterprise application invokes the caller's service subscriptions and answers the call. The enterprise application may request the caller to authenticate himself, such as entering a PIN. For example, the caller may be allowed to purchase items over the telephone and the caller's attributes may have included his credit card number. In this example, the caller is not required to enter his credit card number but the caller is required to

enter a four digit PIN that matches the PIN in his caller's profile.

If the caller's call requires special routing, the enterprise application forwards the call to an appropriate extension number. For example, the caller may have previously spoken to a call attendant and the enterprise application forwards the call to the same call attendant. Otherwise, the enterprise application forwards the call to the next available attendant.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations, and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

Figure 1 is a diagram showing a call recipient's central office including an initiating caller's identification in a telephone call, and sending the initiating caller's identification to a call recipient;

Figure 2 is a diagram showing a name resolution adapter (NRA) providing caller attribute information to an enterprise application whereby the caller attribute information corresponds to an initiating caller;

Figure 3A is a line information database (LIDB) look-up table that includes caller attributes that correspond to an initiating caller's telephone number;

Figure 3B is an initiating caller authorization look-up table that includes sensitive caller authorizations corresponding to call recipients;

Figure 4 is a diagram showing high-level functional blocks that are included in a name resolution adapter (NRA);

Figure 5 is a high-level flowchart showing steps taken in a name resolution adapter (NRA) receiving initiating caller attributes from a service control point, and sending the caller attributes to an enterprise application (EA);

Figure 6 is a flowchart showing steps taken in a name resolution adapter (NRA) generating a line information database (LIDB) request based upon a call recipient contract agreement and initiating caller authorization agreements;

Figure 7 is a flowchart showing steps taken in an initiating caller configuring authorization entries that authorize call recipients to receive sensitive caller data;

Figure 8 is a flowchart showing steps taken in an enterprise application (EA) processing a call using caller attributes it receives from a name resolution adapter (NRA);

Figure 9 is a flowchart showing steps taken in an enterprise application validating a message that corresponds to an incoming call;

Figure 10 is a flowchart showing steps taken in an enterprise application processing a call; and

Figure 11 is a block diagram of an information handling system capable of implementing the present invention.

DETAILED DESCRIPTION

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather, any number of variations may fall within the scope of the invention which is defined in the claims following the description.

Figure 1 is a diagram showing a call recipient's central office including an initiating caller's identification in a telephone call, and sending the initiating caller's identification to a call recipient. Public switched telephone network (PSTN) **100** includes central offices **120** and **140**. Central offices are typically geographically located, such as in a neighborhood or in a business park, and include a "switch" that manages calls to individual customer telephones. Central office **120** includes switch **125** which routes calls to and from initiating caller **105**. Central office **140** includes switch **145** which routes calls to and from enterprise application **190**. PSTN **100** also includes signaling system 7 (SS7) **135** which is the signaling backbone of PSTN **100** in that SS7 passes caller information between central offices.

Initiating caller **105** uses PSTN **100** to place a telephone call to a recipient caller, such as enterprise application **190**. Enterprise application **190** is a telephone system that supports multiple phone lines. For example, imitating caller **105** may wish to call a computer manufacturer's help desk for assistance with configuring his computer.

Initiating caller **105** sends call **110** to central office **120**. Central office **120** includes switch **125** which receives call **110**. Central office **120** performs two functions with call **110**. Its first function is to identify a destination central office that corresponds to call **110**. Central office **120** identifies that call **110** corresponds to a telephone that central office **140** supports. Central office **120** sends call **110** to central office **140** over synchronous optical network (SONET) **128**. Central office **120**'s second function is to send initiating caller **105**'s telephone number (e.g. caller number **130**) to central office **140** through signaling system 7 **135**.

Central office **140** receives and correlates caller number **130** with call **110**. Central office **140** then sends a request (e.g. request **150**) to service control point **160** for a name that corresponds to caller number **130**. Service control point **160** retrieves caller name **175** from line information database (LIDB) store **170**, and sends caller name **175** to central office **140**. Central office **140** includes caller name **175** and caller number **130** in caller identification **180**, and sends caller identification **180** with call **110** to enterprise application **190** over a destination subscriber loop between the first and second ring by means of two modem tones. Caller ID **180** is transmitted serially in FSK mode using either a Single Data Message Format (SDMF) or a Multiple Data Message Format (MDMF). The SDMF includes the date, time, and calling number. The MDMF includes the date, time, calling number, and the name associated with the calling number. Enterprise application **190** uses caller ID **180** to retrieve caller profile information from profile store **195**. Profile

store **195** may be stored on a nonvolatile storage area, such as a computer hard drive.

Figure 2 is a diagram showing a name resolution adapter (NRA) providing caller attribute information to an enterprise application whereby the caller attribute information corresponds to an initiating caller. **Figure 2** is similar to **Figure 1** with regards to sending a call (e.g. call **110**) from initiating caller **105** to enterprise application **190**. However, **Figure 2** is different than **Figure 1** with regards to providing caller information to enterprise application **190**. The diagram in **Figure 2** shows a name resolution adapter retrieving caller attributes, such as name, address, and billing information, and sending the caller attributes to enterprise application **190** over computer network **250**, such as a TCP/IP network.

Central office **140** receives caller number **130** from SS7 **135**, and sends message **210** to name resolution adapter **200**. Message **210** includes caller number **130** as well as the call recipient's (e.g. enterprise application **190**) phone number. Name resolution adapter **200** uses the call recipient's phone number to look-up contract agreement information that is located in contract store **215**. A call recipient sends an agreement request to name resolution adapter **200** in order to establish a contract agreement. For example, enterprise application **190** may have a contract agreement with a name resolution adapter service provider whereby enterprise application **190** requests a name and an address that corresponds with each incoming call. In this example, enterprise application **190** may use an initiating caller's address to route a particular phone call to a company's retail store that is closest to the initiating caller's

address. Contract store **215** may be stored on a nonvolatile storage area, such as a computer hard drive.

Name resolution adapter **200** identifies requested caller fields corresponding to enterprise application **190**'s contract agreement, and includes the requested caller fields in a request (e.g. request **220**) to service control point (SCP) **160**. Service control point **160** receives request **220**, and retrieves caller attributes corresponding to the requested caller fields (e.g. caller attributes **230**) from line information database **170**. Service control point **160** sends caller attributes **230** to name resolution adapter **200**.

Name resolution adapter **200** includes caller attributes **230** in a message as well as telephone port location information that identifies which port enterprise application **190** will receive the corresponding call. Name resolution adapter **200** then sends enterprise application message **240** to enterprise application **190** over computer network **250**, such as a TCP/IP network. Enterprise application **190** validates call **110** using enterprise application message **240**. Enterprise application **190** then uses caller attributes **230** to retrieve caller profile information from profile store **195** to process the call (see **Figures 8, 9, 10,** and corresponding text for further details regarding enterprise application call processing). **Figure 2** shows that enterprise application **190** receives caller attributes **230** and call **110** over heterogeneous, or dissimilar networks. In an alternative embodiment, name resolution adapter **200** may send the caller attributes to central office **140** which, in turn, sends the caller attributes to enterprise application **190** over a telephone

network that is used to send call **110** to enterprise application **190**.

Figure 3A is a line information database (LIDB) look-up table that includes caller attributes that correspond to an initiating caller's telephone number. A service control point accesses table **300** when the service control point receives a request from a name resolution adapter for particular caller information. For example, a name resolution adapter may request an address that corresponds to a particular telephone number (see **Figure 2** and corresponding text for further details regarding NRA requests).

Table **300** includes columns **305** through **325**. Column **305** includes a list of initiating caller telephone numbers. A service control point uses information in column **305** to match a NRA request with a particular initiating caller entry. Column **310** includes a person's name that corresponds to the telephone numbers included in column **305**. Column **315** includes address information that corresponds to initiating caller numbers that are shown in column **305**. Information in columns **310** and **315** are the same as information listed in a telephone directory.

Column **320** includes billing information that corresponds to telephone numbers listed in column **305**. For example, a person may have his phone bill charged to his credit card whereby the person's credit card information is included in column **320**. In this example, a service control point may retrieve billing information from table **300** and provide the billing information to a name resolution adapter which, in turn, sends the billing information to an

enterprise application. In this example, for security purposes, an agreement is in place between the initiating caller and the call recipient that allows the call recipient to receive the initiating caller's billing information. Column **325** includes service preference information corresponding to phone numbers that are included in column **305**, such as call waiting, call forwarding, and three way calling.

Table **300** includes rows **330** through **340** whereby each row includes an initiating caller entry. Row **330** includes caller attributes corresponding to phone number "512-555-1212." The caller attributes include the name "John Doe", the address "800 Anytown Dr.", alternate billing "12324356", and service preferences "call waiting" and "caller id." Row **335** includes caller attributes corresponding to phone number "496-123-4567." The caller attributes include the name "Jane Doe", the address "500 Anystreet", alternate billing "989865447", and service preferences "three way calling." Row **340** includes caller attributes corresponding to phone number "654-987-4321." The caller attributes include the name "Bob Doe", the address "310 Court Ave.", alternate billing "97844675", and service preferences "caller id."

Figure 3B is an initiating caller authorization look-up table that includes sensitive caller authorizations corresponding to call recipients. For example, an initiating caller may wish a mail order catalog to receive the initiating caller's credit card information when the initiating caller places a call to the mail order catalog. In this example, the initiating caller configures an authorization entry at a name resolution adapter service

provider to authorize the name resolution service provider to provide the credit card information to the call recipient.

Table **350** includes columns **360**, **370**, and **375**. Column **360** includes the telephone number of an initiating caller which has configured authorizations corresponding to call recipients (see **Figure 7** and corresponding text for further details regarding call recipient authorization configuration). Column **370** includes call recipient identities (e.g. telephone numbers) that correspond to caller authorizations. As one skilled in the art can appreciate, other call recipient identities may be used to associate a call recipient with a caller authorization, such as the call recipient's name.

Column **375** includes sensitive caller data authorizations corresponding to call recipient identities located in column **370**. Row **380** shows that call recipient "876-543-0989" is authorized to receive "Billing Information" corresponding to initiating caller "512-555-1212." Row **385** shows that call recipient "656-789-6434" is authorized to receive "Birth Date" information corresponding to initiating caller "512-555-1212." And, row **390** shows that call recipient "467-864-2578" is authorized to receive "Social Security Number" information corresponding to initiating caller "512-555-1212."

Figure 4 is a diagram showing high-level functional blocks that are included in a name resolution adapter (NRA). NRA **200** includes PSTN interface **400** which receives message **210** from service termination point **400**. Message **210** includes an initiating caller's number and a call

recipient's number. Message **210** may be formatted in a standard format, such as an Initial Address Message Transaction Capability Application Part (IAM TCAP). PSTN interface **400** passes the initiating caller's phone number and the call recipient's phone number to contract manager **420**. Contract manager **420** looks up contract information corresponding to the call recipient's phone number in contract store **215**. For example, NRA **200** may have a contract agreement with a call recipient to provide an initiating caller's address with each incoming telephone call.

Contract manager **420** retrieves requested caller fields from contract store **215**, and contract manager **420** passes requested caller fields **425** to line information database (LIDB) message manager **430**. LIDB message manager **430** includes requested caller fields **425** in request **220**, and sends request **220** to service control point (SCP) **160**. Request **220** may be formatted in a standard format, such as TCAP. SCP **160** retrieves caller attributes from LIDB store **170**, and sends caller attributes **230** to LIDB message manager **430**. LIDB message manager **430** forwards caller attributes **230** to enterprise application message manager **440**.

Enterprise application message manager **440** includes caller attributes **230** in EA message **240**, along with a telephone port location that identifies the port at which enterprise application **190** receives the corresponding call. Enterprise application message manager then sends EA message **240** to enterprise application **190** over a TCP/IP network connection.

Figure 5 is a high-level flowchart showing steps taken in a name resolution adapter (NRA) receiving initiating caller attributes from a service control point, and sending the caller attributes to an enterprise application (EA). Processing commences at **500**, whereupon the NRA waits for a call from initiating caller **105** through PSTN **100** at step **510**. Initiating caller **105** and PSTN **100** are the same as that shown in **Figure 1**. When the NRA receives a call, the NRA identifies a recipient of initiating caller **105**'s telephone call at step **520**. For example, initiating caller **105** may be calling a company's customer support line.

At step **530**, NRA processing identifies a contract agreement that corresponds to the call recipient, retrieves requested caller fields from contract store **215**, and generates a line information database (LIDB) request using the requested caller fields (pre-defined process block, see **Figure 6** and corresponding text for further details). NRA processing sends the LIDB request to service control point **160** at step **540**. In turn, NRA processing receives caller attributes corresponding to the LIDB request from service control point **160**, and stores the caller attributes in temp store **555** (step **550**).

NRA processing includes the caller attributes received from temporary store **555** and caller fields from contract store **215** in an enterprise application (EA) message at step **560**. Processing sends the EA message to enterprise application **190** through computer network **250** at step **570**. EA application **190** receives the EA message, and associates the caller attributes with initiating caller **105**'s telephone call (see **Figures 2**, **8**, and corresponding text

for further details regarding enterprise application call processing).

A determination is made as to whether to continue processing telephone calls (decision **580**). If processing should continue, decision **580** branches to "Yes" branch **582** which loops back to process another call. This looping continues until NRA call processing should stop, at which point decision **580** branches to "No" branch **588** whereupon processing ends at **590**.

Figure 6 is a flowchart showing steps taken in a name resolution adapter (NRA) generating a line information database (LIDB) request based upon a call recipient contract agreement and initiating caller authorization agreements. LIDB request generation processing commences at **600**, whereupon processing identifies at **610** whether the call recipient has a contract agreement with the NRA service provider by accessing a contract look-up table located in contract store **215**. For example, a company may have a contract with an NRA service provider whereby the service provider provides an initiating caller's name and address with each telephone call.

A determination is made as to whether the call recipient has an existing contract agreement (decision **620**). If the call recipient does not have an existing contract agreement, decision **620** branches to "No" branch **622** whereby processing logs call information in log store **628** at step **625**. In one embodiment, the name resolution adapter increments a counter to track the number of times that it receives a message corresponding to a particular call recipient. The name resolution adapter service

provider may use the counter and logged information to show a potential customer the number of calls the potential customer could receive that would include caller attribute information. Log store **628** may be stored on a nonvolatile storage area, such as a computer hard drive. Processing returns at **629**.

On the other hand, if the call recipient has an existing contract agreement with the NRA service provider, decision **620** branches to "Yes" branch **623** whereupon at step **630** processing looks up requested caller fields corresponding to the contract agreement located in contract store **215**. Contract store **215** is the same as that shown in **Figure 2** and may be stored on a nonvolatile storage area, such as a computer hard drive.

At **640**, a determination is made as to whether one or more of the call recipient's requested caller fields correspond to sensitive caller data. For example, a call recipient may request the initiating caller's credit card number. If the call recipient does not request sensitive caller data, decision **640** branches to "No" branch **642** whereupon processing includes each requested caller field in request **220** at step **645**, and processing returns at **650**.

On the other hand, if one or more of the call recipient's requested caller fields corresponds to sensitive caller data, decision **640** branches to "Yes" branch **644** whereupon processing looks up initiating caller authorization agreements in contract store **215** (step **665**). Using the example described above, an initiating caller may authorize the call recipient to authorize the call recipient to receive the initiating caller's credit card

number (see **Figure 3B** and corresponding text for further details regarding initiating caller authorization details).

A determination is made as to whether the initiating caller authorizes the call recipient to receive sensitive caller data (decision **660**). If the call recipient is authorized to receive the initiating caller's sensitive caller data, decision **660** branches to "Yes" branch **662** whereupon processing includes each requested caller field in request **220** (step **645**). On the other hand, if the call recipient is not authorized to receive sensitive caller data, decision **660** branches to "No" branch **668** whereupon processing logs the discrepancy in log store **628** in order to provide the discrepancy information to a call recipient. Processing includes only authorized requested caller fields in request **220** at step **680**, and processing returns at **690**.

Figure 7 is a flowchart showing steps taken in an initiating caller configuring authorization entries that authorize call recipients to receive sensitive caller data. For example, an initiating caller may wish a mail order catalog to receive the initiating caller's credit card information when the initiating caller places a call to the mail order catalog. In this example, the initiating caller configures an authorization entry at a name resolution adapter service provider to authorize the name resolution service provider to provide the credit card information to the call recipient.

Processing commences at **700**, whereupon processing receives an authorization setup request from initiating caller **105** (step **710**). Initiating caller **105** is the same as that shown in **Figure 1**. At step **720**, processing

generates a caller authorization entry in an authorization look-up table stored in contract store **215** (see **Figure 3B** and corresponding text for further details regarding authorization table properties). Contract store **215** is the same as that shown in **Figure 2**.

Processing receives a first call recipient identity from initiating caller **105**, such as the call recipient's phone number, at step **730**, and stores the call recipient's identity in the authorization look-up table located in contract store **215** (step **740**). Processing then receives a first authorization (i.e. credit card authorization) to correspond to the first call recipient identity at step **750**, and stores the first authorization in the authorization look-up table stored in contract store **215** (step **760**).

A determination is made as to whether initiating caller **105** wishes to add more authorizations to correspond to the first call recipient (decision **770**). For example, initiating caller **105** may wish to authorize the first call recipient to receive the initiating caller's social security number. If initiating caller **105** wishes to add more authorizations to correspond to the first call recipient, decision **770** branches to "Yes" branch **772** whereupon processing receives (step **775**) and processes the next authorization to correspond to the first call recipient. This looping continues until initiating caller **105** does not wish to add more authorizations to the first call recipient, at which point decision **770** branches to "No" branch **778** whereupon a determination is made as to whether initiating caller **105** wishes to configure authorizations for additional call recipients (decision

780). If initiating caller wishes to configure authorizations for additional call recipients, decision **780** branches to "Yes" branch **782** whereupon processing receives (step **785**) and processes a second call recipient's authorizations. This looping continues until there are no more call recipients to process, at which point decision **780** branches to "No" branch **788** whereupon processing ends at **790**.

Figure 8 is a flowchart showing steps taken in an enterprise application (EA) processing a call using caller attributes it receives from a name resolution adapter (NRA). Enterprise application processing commences at **800**, whereupon the enterprise application waits for a message from name resolution adapter **200** through computer network **250** (step **810**). The message includes one or more customer identifiers, such as caller attributes, and corresponds to a call from an initiating caller. Name resolution adapter **200** and computer network **250** are the same as those shown in **Figure 2**.

The enterprise application validates the call using the caller attributes and also verifies that the call is present on a specified telephone port (pre-defined process block **830**, see **Figure 9** and corresponding text for further details). For example, a company's customer support telephone system may identify that caller is calling from a long distance telephone based upon caller attributes and decides not to accept the call in order to avoid long distance charges on its 1-800 telephone line.

A determination is made as to whether the call corresponding to the message is valid based upon profile attributes (decision **840**). If the call is not valid,

decision **840** branches to "No" branch **842** whereupon at step **850** processing sends an error message to name resolution adapter **200** which, in turn, provides the error message to the caller. For example, processing may play a recording to the caller as to which number to dial based upon the caller's address.

On the other hand, if the call is valid, decision **840** branches to "Yes" branch **848** whereupon a determination is made as to whether the telephone port corresponding to the message is receiving a call (decision **860**). If the port corresponding to the message is not receiving an incoming call, decision **860** branches to "No" branch **862** which loops back to log the call discrepancy in unassociated call store **875** (step **870**), and wait for another message. This looping continues until a telephone port corresponding to a message includes an incoming call, at which point decision **860** branches to "Yes" branch **868** whereupon the enterprise application processes the call (pre-defined process block **880**, see **Figure 10** and corresponding text for further details).

A determination is made as to whether to continue processing incoming messages (decision **890**). If processing should continue to process incoming messages, decision **890** branches to "Yes" branch **892** which loops back to process more messages. This looping continues until the enterprise application does not wish to process more incoming messages, at which point decision **890** branches to "No" branch **898** whereupon processing ends at **899**.

Figure 9 is a flowchart showing steps taken in an enterprise application validating a telephone call that corresponds to an incoming message. Processing commences at **900**, whereupon processing extracts caller attributes

from enterprise application message **240** at step **910**. Enterprise message **240** is the same as that shown in **Figure 2**. At step **920**, processing uses the caller attributes, such as the caller's name, to look-up a caller profile in profile store **195**. For example, the enterprise application may support a banking telephone system whereby the enterprise application looks-up the caller's account information. Profile store **195** is the same as that shown in **Figure 1**.

A determination is made as to whether to accept the call based upon the caller's profile information (decision **930**). For example, if the caller does not have an account with the bank, the enterprise application may not accept the call. If the enterprise application should not accept the call, decision **930** branches to "No" branch **932** whereupon processing returns an error at **935**. On the other hand, if processing should accept the call, decision **930** branches to "Yes" branch **938** whereupon processing extracts telephone port location information from enterprise application **240** at step **940**. The telephone port location information corresponds to which port (i.e. circuit) the enterprise application should receive the corresponding call.

Processing checks the identified port location for an incoming call at step **950**, and a determination is made as to whether a call is present at the specified port location (decision **960**). If a call is present at the specified port location, decision **960** branches to "Yes" branch **968** whereupon processing returns at **990**. If a call is not present at the specified port location, decision **960** branches to "No" branch **962** whereupon a determination is made as to whether processing should wait for the incoming

call (decision **970**). For example, processing may set a timer for two seconds whereby processing waits for two seconds after it receives a message to look for a corresponding call.

If processing should wait for the incoming call, decision **970** branches to "No" branch **972** whereupon processing loops back to check for the corresponding call at the specified port location. This looping continues until processing's timer times out, at which point decision **970** branches to "Yes" branch **978** whereupon processing returns an "unassociated call" at **980**.

Figure 10 is a flowchart showing steps taken in an enterprise application processing a call. Call processing commences at **1000**, whereupon processing retrieves subscriptions corresponding to the caller from subscriptions store **1005** (step **1010**). For example, processing may retrieve a level of service for a caller that permits him to check billing information, but his level of service does not permit him to buy anything. Processing invokes a service corresponding to the service subscriptions at step **1020**. Using the example described above, processing may configure permissions and associate the permissions to the call which allows the caller to check billing information but not buy anything.

The enterprise application answers the call at step **1030**, whereupon a determination is made as to whether to validate the caller (decision **1040**). For example, a banking application may require a user to enter a PIN in order to authenticate the user. If processing does not need to validate the caller, decision **1040** branches to "No" branch **1048** bypassing caller validation steps. On the other hand, if processing should validate the caller,

decision **1040** branches to "Yes" branch **1042** whereupon processing requests a PIN from initiating caller **105** at step **1045**. Initiating caller **105** is the same as that shown in **Figure 1**.

Processing receives a PIN from initiating caller **105** at step **1050**, and a determination is made as to whether initiating caller **105** entered the correct PIN (decision **1055**). If initiating caller **105** did not enter the correct PIN, decision **1055** branches to "No" branch **1057** whereupon processing plays an error message to initiating caller **105** at step **1060**, and processing returns to receive more calls at **1065** (see **Figure 8** and corresponding text for further details). On the other hand, if initiating caller **105** did enter the correct PIN, decision **1055** branches to "Yes" branch **1059**.

A determination is made as to whether the caller requires special routing to a particular extension (decision **1070**). For example, a caller may have previously spoken to a particular attendant and the enterprise application wishes to route the caller's call to the same attendant. If the call requires special routing, decision **1070** branches to "Yes" branch **1078** whereupon processing forwards the call to a particular number (step **1085**), such as attendant **1090**'s number, and processing returns to receive more calls at **1095** (see **Figure 8** and corresponding text for further details). On the other hand, if the call does not require special routing, decision **1070** branches to "No" branch **1072** whereupon processing routes the call to the next attendant (step **1075**), such as attendant **1080**'s number, and processing returns to receive more calls at **1095** (see **Figure 8** and corresponding text for further details).

Figure 11 illustrates information handling system **1101** which is a simplified example of a computer system capable of performing the computing operations described herein. Computer system **1101** includes processor **1100** which is coupled to host bus **1102**. A level two (**L2**) cache memory **1104** is also coupled to host bus **1102**. Host-to-PCI bridge **1106** is coupled to main memory **1108**, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus **1110**, processor **1100**, L2 cache **1104**, main memory **1108**, and host bus **1102**. Main memory **1108** is coupled to Host-to-PCI bridge **1106** as well as host bus **1102**. Devices used solely by host processor(s) **1100**, such as LAN card **1130**, are coupled to PCI bus **1110**. Service Processor Interface and ISA Access Pass-through **1112** provides an interface between PCI bus **1110** and PCI bus **1114**. In this manner, PCI bus **1114** is insulated from PCI bus **1110**. Devices, such as flash memory **1118**, are coupled to PCI bus **1114**. In one implementation, flash memory **1118** includes BIOS code that incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions.

PCI bus **1114** provides an interface for a variety of devices that are shared by host processor(s) **1100** and Service Processor **1116** including, for example, flash memory **1118**. PCI-to-ISA bridge **1135** provides bus control to handle transfers between PCI bus **1114** and ISA bus **1140**, universal serial bus (USB) functionality **1145**, power management functionality **1155**, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus support. Nonvolatile RAM **1120** is attached to ISA Bus **1140**. Service Processor **1116** includes JTAG and

I2C busses **1122** for communication with processor(s) **1100** during initialization steps. JTAG/I2C busses **1122** are also coupled to L2 cache **1104**, Host-to-PCI bridge **1106**, and main memory **1108** providing a communications path between the processor, the Service Processor, the L2 cache, the Host-to-PCI bridge, and the main memory. Service Processor **1116** also has access to system power resources for powering down information handling device **1101**.

Peripheral devices and input/output (I/O) devices can be attached to various interfaces (e.g., parallel interface **1162**, serial interface **1164**, keyboard interface **1168**, and mouse interface **1170** coupled to ISA bus **1140**. Alternatively, many I/O devices can be accommodated by a super I/O controller (not shown) attached to ISA bus **1140**.

In order to attach computer system **1101** to another computer system to copy files over a network, LAN card **1130** is coupled to PCI bus **1110**. Similarly, to connect computer system **1101** to an ISP to connect to the Internet using a telephone line connection, modem **1175** is connected to serial port **1164** and PCI-to-ISA Bridge **1135**.

While the computer system described in **Figure 11** is capable of executing the processes described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

One of the preferred implementations of the invention is an application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until

required by the computer, the set of instructions may be stored in another computer memory, for example, on a hard disk drive, or in removable storage such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer program product for use in a computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For a non-limiting example, as an aid to understanding, the following appended claims contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of

such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use in the claims of definite articles.